

# OPTIMIZATION OF ROBOT'S TRAJECTORY

Miroslav ŠTĚPÁN, Master Degree Programme (4)  
Dept. of Intelligent Systems, FIT, BUT  
E-mail: xstepa22@stud.fit.vutbr.cz

Supervised by: Ing. Martin Hrubý

## ABSTRACT

This project deals with possible algorithms for autonomous intelligent movement which could be used for controlling the robot in unknown environment.

## 1 ÚVOD

Smyslem této práce je specifikace metod pro samostatný inteligentní pohyb robota v neznámém 2D prostředí. Inteligentním pohybem se myslí zejména schopnost efektivního obcházení překážek a plynulost pohybu. Pokud tedy celkové chování systému rozdělíme na dvě hlavní metody, jsou jimi rozhodování a optimalizace pohybu.

## 2 IMPLEMENTACE

Implementace na skutečném robotu je závislá jak na vlastnostech jeho senzorů, tak i na jeho pohybových schopnostech. Reálné senzory mají omezený dosah a malé rozlišovací schopnosti. Pohyb se nedá ideálně kontrolovat, není plynulý atd. Proto by ideální HW platforma na které by se výsledný systém měl implementovat musela splňovat mnoho požadavků které většina dostupných prostředků nenabízí. Ideální robot by musel mít senzory s velkým dosahem, rozsahem a velkou rozlišovací schopností. Dokázal by kontrolovat svůj pohyb s dostatečnou přesností a dokázal by plynule měnit směr i rychlost.

Protože hlavním cílem je konstrukce algoritmů pro rozhodování a optimalizaci pohybu, je vhodnější přesunout pozornost na tyto metody, než na rozhraní mezi SW a HW které by muselo řešit zmíněné nedostatky HW. Proto je vhodné systém implementovat na SW modelu robota simulujícího požadovaný sensorický vstup i reakce na podněty systému. Celá simulovaná platforma se poté dá jednoduše parametrizovat a systém se tak dá otestovat na různé úrovni kvality vstupů atd.

Pro začátek je samozřejmě vhodné se nad tyto problémy co nejvíce abstrahovat. Budeme tedy předpokládat, že model robota odpovídá ideální platformě.

Dalším problémem je rychlost systému – měla by být taková aby dovozovala zpracování vstupů v reálném čase, protože případné prodlevy by měly negativní vliv na podstatu optimalizace – na plynulost pohybu. Tento problém však vyvstává až při implementaci na

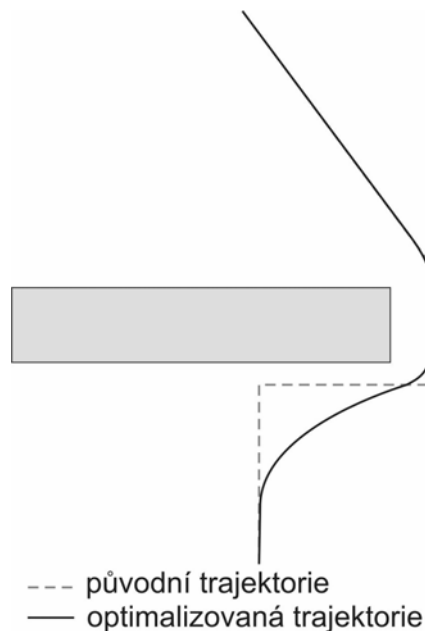
skutečném robotu. Při simulaci však dojde pouze k prodloužení simulačního času.

### 3 POPIS VLASTNÍCH ALGORITMŮ

#### 3.1 ROZHODOVÁNÍ

Robot se pohybuje po trajektorii, která je primárně určena úsečkou mezi výchozím místem a cílem. Pokud při jeho postupu senzory zaznamenají v cestě nějakou překážku, musí jí obejít.

Nejdříve musí vyhodnotit relevantní části hranice dané překážky. Při tom postupuje směrem od průsečíku jeho původní trajektorie do stran. Nejbližší místo označené jako „hrana“ překážky, resp. bod který je od tohoto místa vzdálen o minimální bezpečnou vzdálenost, je dalším (sekundárním) cílem. Tímto způsobem robot postupuje dokud nedosáhne primárního cíle.



**Obr. 1:** ilustrace optimalizace trajektorie

#### 3.2 OPTIMALIZACE

Optimalizace trajektorie s ohledem na plynulost je vlastně minimalizace okamžitých změn v pohybu robota. Jinými slovy, jde o vyhlazení hran původní trajektorie. Je však otázkou do jaké míry je vhodné pohyb optimalizovat. S větší mírou plynulosti dochází k prodlužování dráhy a tím snižováním efektivity původně inteligentní cesty. Míra optimalizace je tedy zásadním parametrem ovlivňujícím pohyb robota.

Aby pohyb byl plynulý, musí se veškeré změny směru i rychlosti rozložit na co největší prostor. Ale vzhledem k tomu, že dosah senzorů je omezený, je omezená i možnost optimalizace pohybu. Zde je základní rozpor mezi optimalizací pohybu a rozhodováním robota. Není totiž možné aby robot inteligentně určil svoji trajektorii ihned když „vidí“ pouze část překážky která mu stojí v cestě. Je potřeba postupovat dál v původní trajektorii dokud

v zorném poli nemá takovou část překážky, podle které je schopen určit směr dalšího postupu. A tím se zkracuje prostor pro optimalizaci pohybu. Pokud není pro změnu směru, potažmo rychlosti dostatek místa, působí změna neplynule. Krajním případem důsledku malého dosahu senzorů vůči velikosti překážky je kopírování hranice překážky.

Pomocným nástrojem optimalizace může být například predikce tvaru překážky – pokud robot zjistí v dosahu senzorů nějakou překážku (ne přímo překážku ve směru předpokládaného postupu), může podle pozice a tvaru překážky v „zorném poli“ a parametrů tohoto pole odhadnout zda překážka může stát skutečně v cestě i když tam jí zatím není možné detekovat. Tato predikce může pomoci zrychlit „reakční dobu“ při obcházení překážky a zvýšit tak plynulost trajektorie, zvláště při malém dosahu senzorů vzhledem k velikosti překážky. Tato metoda může řešit malý dosah senzorů, ale nemá smysl, pokud je malý směrový rozsah senzorů. Protože je poměrně náročná, není vhodné ji používat v prostředí které nevyžaduje velké výchylky z trasy (např. pokud jsou překážky dostatečně malé).

#### 4 TECHNICKÉ ŘEŠENÍ

Jako vstup „simulátoru“ jsem použil bitmapu, která představuje prostředí v kterém se robot bude pohybovat. Hodnota jednotlivých bodů bitmapy určuje vlastnosti prostředí v odpovídajících místech prostředí. V nejjednodušším případě jsou to jen hodnoty označující přítomnost překážky nebo volné místo. Dále je možné v bitmapě zaznamenat výchozí pozici robota a cíle. Při inicializaci prostředí se načte bitmapa a pokud v ní jsou označeny tyto místa, určí se jejich pozice.

Funkcí tohoto prostředí je simulace senzorických vstupů na základě pozice a natočení robota a parametrů jeho senzorů. V podstatě je to funkce která namapuje body z bitmapy na základě těchto parametrů do „zorného pole“ robota. To je představováno polem hodnot určujících podobné vlastnosti jako hodnoty bodů bitmapy. Opět v nejjednodušší definici prostředí to odpovídá hodnotám typu „vidím překážku“, „vidím volno“ nebo „nevidím nic“.

System řízení poté spočívá v aplikaci výše popsaných algoritmů pro určování a optimalizaci trajektorie na „zorné pole“ robota. Jejich výstupem je pouze nově určená poloha, která se předává na vstup simulátoru prostředí.

#### 5 ZÁVĚR

Toto řešení inteligentního pohybu robota je pro praktické použití značně omezené. Předpokládáme 2D prostředí kde rozeznáváme pouze jeden typ povrchu a přítomnost překážky. Nabízí se tedy zpřesnění parametrů prostředí na reálnější úroveň – více hodnot specifikujících prostředí (různé vlastnosti povrchů, charakter překážky). Tento krok už by vyžadoval náročnější způsoby vyhodnocování a prohledávání stavového prostoru pro určování trajektorie.

#### LITERATURA

- [1] Hrubý M.: studijní materiály pro kurz Modelování a simulace – FIT, VUT Brno <http://www.fit.vutbr.cz/study/courses/MSI>
- [2] Zbořil F.: studijní materiály pro kurz Umělá inteligence – FIT, VUT Brno <http://www.fit.vutbr.cz/study/courses/UIN>